

ISPIT 1

1. **Općenito, za funkcionalnu jedinicu programskog sustava koja se naziva komponenta vrijedi da se može neovisno postavljati i nadograđivati.**
2. **Kod praktične primjene filozofije u kojoj se sva programska logika izmješta iz posredničkog sloja, sigurno se ne koristi:**
Enterprise Service Bus (ESB).
3. **U usporedbi s jediničnim (unit) testovima, testovi krajnjih točaka su**
sporiji i općenito ih ima manje.
4. **Ono što je WSDL u SOA tehnološkom stogu, to je za sustav gRPC**
protocol buffers.
5. **Glavna motivacija za korištenje sustava Apache Thrift je**
učinkovita i pouzdana komunikacija između komponenti ostvarenih primjenom različitih protokola i transportnih medija.
6. **Prikazani isječak koda napisan je u jeziku za opis sučelja (IDL) sustava**

```
message Result {  
    required string url = 1;  
    optional string title = 2;  
    repeated string snippets = 3;  
}
```

Apache Thrift.

7. **Glavna prednost protokola WebSocket je**
upućivanje HTTP zahtjeva prema poslužitelju bez čekanja HTTP odgovora na prethodno upućeni zahtjev.
8. **U okviru standardnog WebSocket programskog sučelja (engl. WebSocket API), čitanje poruke na prijemnoj strani izvodi se**
registracijom korisničke funkcije onmessage koja se poziva mehanizmom povratnog poziva.
9. **U sustavu s pravilnim periodičkim promjenama gdje je period promjene dugačak (primjerice, svakih nekoliko sati), a količina korisnih podataka značajno veća od zaglavlja HTTP protokola, promjene s poslužitelja na klijente preporučljivo je izvesti**
tehnikom prozivanja (polling).
10. **Zadane su dvije inačice jedno te iste web usluge s informacijama o predmetu na studijskom programu. Informacije o predmetu sastoje se od: 1) općih podataka o predmetu, 2) podataka o nastavnicima na predmetu i 3) podataka o literaturi. Inačice se razlikuju po načinu organizacije ponuđenih informacija u sredstva poslužitelja. U tablicama su zadane web adrese pojedinih sredstava i veličine tih sredstava za dvije inačice web usluge. Web usluzi pristupa ukupno 1000 studenata na sljedeći način:**
 - a. 100 studenata treba sve informacije o predmetu
 - b. 100 studenata treba samo informacije o nastavnicima, ali svima koji sudjeluju u izvođenju predmeta

- c. 600 studenata treba informaciju o samo jednom nastavniku, onom u čijoj grupi polazi nastavu
- d. 200 studenata treba samo informacije o literaturi.

Veličina zaglavlja svakog HTTP zahtjeva i svakog HTTP odgovora je 500 okteta (zajedno s praznim retkom na kraju zaglavlja).

Inačica A		Inačica B	
URL sredstva	Veličina sredstva	URL sredstva	Veličina sredstva
/predmet/osnovno	100	/predmet	3600
/predmet/nastavnici/nastavnik1	1000	/predmet/nastavnici	3000
/predmet/nastavnici/nastavnik2	1000	/predmet/literatura	500
/predmet/nastavnici/nastavnik3	1000		
/predmet/literatura	500		

11. U web usluzi za čavrljanje (chat) nalaze se dva klijenta (klijent 1 i klijent 2) koji razmjenjuju tekstualne poruke putem posredničkog poslužitelja. Klijent pošiljatelj poruku šalje poslužitelju, a poslužitelj ju nakon toga šalje klijentu primatelju. Oba klijenta mogu poprimiti ulogu pošiljatelja i primatelja. Dinamika slanja poruka s klijenata pošiljatelja prikazana je u tablici. Vrijeme potrebno za prijenos poruka mrežom, kao i za obradu na poslužitelju i na klijentima je zanemarivo. Vrijeme teče od trenutka 0 i odmah nakon toga klijenti i poslužitelj su spremni za rad. Poslužitelj za svakog klijenta pamti samo zadnju poruku koju je taj klijent poslao. Odredite vremena isporuke poruka klijentima primateljima ako se za isporuku poruka koristi

- a. prozivanje (polling), pri čemu klijent 1 radi s periodom od 5 sekundi, a klijent 2 s periodom od 10 sekundi
- b. blokirajuće prozivanje (long polling)
- c. protokol WebSocket.

Odgovore upisati u tablicu.

ISPIT 2

- Preporučeni način za izvedbu potpore udomljavanju web sredstava na zajedničkoj poslužiteljskoj infrastrukturi (engl. Virtual web hosting) je:**
raspoznavanje usluga po Host zaglavlju u HTTP zahtjevu.
- Cjevovodni režim rada HTTP protokola (engl. HTTP pipelining) omogućava**
upućivanje HTTP zahtjeva prema poslužitelju bez čekanja HTTP odgovora na prethodno upućeni zahtjev.
- Ispravno oblikovanim REST sučeljem (engl. REST api) smatra se ono koje je klijentima izloženo na način**
DELETE /users/1234
- Oblikovanje URL adresa za dva sredstva X i Y koji su u arhitekturnom stilu REST na istoj razini hijerarhije preporučljivo je izvesti na način**
<http://api.example.com/x> za sredstvo X, <http://api.example.com/y> za sredstvo Y
- X na višoj razini, Y na nižoj razini**
<http://api.example.com/x> za sredstvo X, <http://api.example.com/x/y> za sredstvo Y
- U sustavu s pravilnim periodičnim promjenama gdje je period promjene dugačak (primjerice svakih nekoliko sati), a količina korisnih podataka značajno veća od zaglavlja HTTP protokola, promjene s poslužitelja na klijente preporučljivo je izvesti**

tehnikom prozivanja (polling).

7. **Nakon početne uspostave veze, tehnologija SSE omogućava komunikaciju klijenta i poslužitelja koja je**
jednosmjerna od poslužitelja prema klijentu.
8. **Nakon početne uspostave veze, protokol WebSocket omogućava komunikaciju klijenta i poslužitelja koja je**
dvosmjerna asinkrona.
9. **Protokolom WebSocket moguće je prenositi**
tekstualne i binarne poruke.
10. **Protokolom SSE moguće je prenositi**
tekstualne izravno, binarne samo ako su prethodno kodirane u Base64 format.
11. **U okviru standardnog WebSocket programskog sučelja čitanje poruka na prijemnoj strani izvodi se**
registracijom korisničke funkcije na događaj onmessage koja se poziva mehanizmom povratnog poziva.
12. **Mehanizam integriranih usluga (IntServ) zasniva se na**
unaprijednoj rezervaciji sredstava.
13. **Mehanizam razlikovnih usluga (DiffServ) zasniva se na**
prioritetnom posluživanju s obzirom na vrstu mrežnog prometa.
14. **Glavna zadaća RPC sustava je**
normalizacija semantike poziva funkcija u istim i različitim adresnim prostorima.
15. **RPC tehnologija zasnovana na jeziku XML bez službenog IDL prevoditelja je:**
XML-RPC.

<pre>var p={ "firstName":"John", "lastName":"Smith" }; var r=p.firstName;</pre>	<pre>var p='{ "firstName":"John", "lastName":"Smith" }'; var r=p.firstName;</pre>
--	--

16.

- a. **Objasniti razliku između lijevog i desnog koda.**

U lijevom kodu varijabla p predstavlja JSON objekt koji se sastoji od atributa firstName i lastName, te će se naredba var r = p.firstName; uspješno izvršiti i rezultirati spremanjem stringa 'John' u varijablu r. U desnom kodu varijabla p predstavlja string koji izgleda kao JSON objekt ali se neće tako protumačiti zato što je napisan u obliku znakovnog niza. Naredba var r = p.firstName; neće se uspješno izvršiti i bacit će error. Kako bi se izbjeglo bacanje exceptiona u desnom kodu, potrebno je parsirati string u JSON objekt korištenjem naredbe var r = (JSON.parse(p)).firstName;

- b. **Uvesti minimalne potrebne dopune na kod kako bi se isječci kodova izvršili ispravno na JavaScript interpretatoru.**

LIJEVI KOD

```
var p = {  
    „firstName“:“John“,  
    „lastName“:“Smith“  
};  
var r = p.firstName;
```

DESNI KOD

```
var p = '{  
    „firstName“:“John“,  
    „lastName“:“Smith“  
}';  
var r = (JSON.parse(p)).firstName;
```

17. Svaki dio 1 bod.

- a. **Navesti barem 3 binarna protokola za RPC.**
Protocol Buffers, Apache Thrift, gRPC.
- b. **Navesti barem 3 tekstualna protokola za RPC.**
XML-RPC, JSON-RPC, SOAP.

18. Opisati oblik i nazivlje u zapisu semantičkog verzioniranja te objasniti što znači promjena pojedinačne stavke u zapisu.

MAJOR.MINOR.PATCH

MAJOR -> Ova verzija se mijenja kada se objavljuju promjene u softveru koje nisu unatrag kompatibilne. Takve vrste promjena zahtijevaju prilagodbe kod korištenja nove programske potpore.

MINOR -> Ova verzija se mijenja kada se objavljuju nove funkcionalnosti, ali postojeće funkcionalnosti ostaju nepromijenjene, odnosno promjene su unatrag kompatibilne. Takve vrste promjena ne zahtijevaju prilagodbe kod korištenja nove programske potpore.

PATCH -> Ova verzija se mijenja kada se objavljuju manje ispravke i sigurnosni popravci. Takve vrste promjena ne utječu na funkcionalnost i ne zahtijevaju prilagodbe kod korištenja nove programske potpore.

19. Navedite prednosti i nedostatke postavljanja svake mikrousluge na zaseban fizički poslužitelj.

Prednosti: nema Single Point Of Failure (ako samo jedan dio sustava zakaže, zaustavit će se rad cijelog sustava), bolja i lakša skalabilnost, alternativne tehnike razvoja programske potpore.

Nedostaci: veća cijena, dodatna kompleksnost.

20. U nadzornom sustavu tunela poslužitelj prikuplja informacije o odvijanju prometa i periodički ih dojavljuje klijentima. Svaka dojava obavlja se posebno oblikovanom tekstualnom porukom u koju su ugrađene prometne informacije. Struktura poruke unaprijed je poznata klijentima i poslužitelju i uvijek je iste veličine, a iznosi 1000 okteta. Prijenos poruka s prometnim informacijama ostvaren je tehnologijom Server-Sent Events. Operater tunela dobio je ponudu za modernizaciju dojavnog dijela sustava nadogradnjom na suvremeniji WebSocket protokol, dok bi svi ostali parametri rada sustava ostali isti. On je sada u dilemi isplati li mu se ulaganje u modernizaciju pa je zatražio konzultantske usluge FER-a. Konzultantima na FER-u je rekao kako mu je cilj optimirati rad sustava s obzirom na količinu ukupno generiranog mrežnog prometa i želi znati pod kojim uvjetima je moguće postići uštedu od 1 GB (2^{30} okteta) mrežnog prometa. Odredite okolnosti u kojima se to postiže s obzirom na parametre rada sustava zadane tablicom. Pretpostavite da je veličina zaglavlja svakog HTTP zahtjeva i svakog HTTP odgovora po 500 okteta zajedno s praznim retkom na kraju zaglavlja. Oznaka zaglavlja sama za sebe zahtijeva 2 okteta ($\backslash r \backslash n$). U trenutnoj inačici sustava koja koristi Server-Sent Events koristi se najjednostavniji oblik dojave koja sadrži samo

redak s oznakom data: nakon čega odmah slijede korisni podaci (bez razmaka). Protokol WebSocket koristio bi standardom definiranu strukturu WebSocket okvira (WebSocket frame).

Bit	+0..7			+8..15		+16..23	+24..31
0	FIN		Opcode	Mask	Length	Extended length (0–8 bytes) ...	
32	...						
64	...					Masking key (0–4 bytes) ...	
96	...					Payload ...	
...	...						

Zadatak	Broj klijenata		Period slanja poruke	Vrijeme do isplativosti (u satima)		
16.	1 000		1 s	A) 0,0596 D) 59,65	B) 298,26 E) 0,5965	C) 59 652
17.	100 000		1 s	A) 0,0596 D) 59,65	B) 2,98 E) 0,5965	C) 59 652
18.	1 000 000		1 s	A) 0,0596 D) 59,65	B) 0,298 E) 0,5965	C) 59 652
19.	1 000		10 s	A) 596,52 D) 0,596	B) 5,965 E) 2982,61	C) 596 523
20.	100 000		10 s	A) 596,52 D) 0,596	B) 5,965 E) 29,82	C) 596 523
21.	1 000 000		10 s	A) 596,52 D) 0,596	B) 5,965 E) 2,982	C) 596 523
22.	1 000		60 s	A) 3 579 139 D) 3,579	B) 3 579,139 E) 17 895,69	C) 35,79
23.	100 000		60 s	A) 3 579 139 D) 3,579	B) 3 579,139 E) 178,95	C) 35,79
24.	1 000 000		60 s	A) 3 579 139 D) 3,579	B) 3 579,139 E) 17,89	C) 35,79
25.	A) 62 138 C) 24 855	B) 124 276 D) 24 856 E) 41 426	10 s	24		

HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 3
Expires: [vrijednost iz tablice 1]
[troznamenkasta numerička vrijednost sredstva]

Tablica 1.

Trenutak promjene sredstva	0	7	13	28	31	37	52	58	93	107	110
Vrijednost sredstva	111	222	333	444	555	666	777	888	999	111	222
Vrijednost Expires zaglavlja	10	17	23	38	41	47	62	68	103	117	120

Tablica 2.

Trenutak slanja zahtjeva s klijenta	2	11	16	19	38	42	55	56	57	60	61
Primljena vrijednost sredstva na klijentu	111	222	222	333	444	555	777	777	777	999	222
Način obrade zahtjeva na posredniku	CH	RH	PH	CH	CH	PH	CH	CH	CH	CH	CH

21.

- Odrediti vrijednost sredstva koju klijent dobiva za svaki poslani zahtjev. Odgovore upisati u tablicu 2.
- Za svaki zahtjev odrediti način njegove obrade na posredniku. Odgovore upisati u tablicu 2 na način: CH = cache hit, CM = cache miss, RH = revalidate hit, RM = revalidate miss

- c. **Odrediti učinkovitost privremene memorije posrednika (engl. hit rate) pod pretpostavkom (odgovore upisati na crtu):**
 - i. da se pogotkom smatra samo CH
 - ii. da se promašajem smatra samo CM
- d. **Komentirati ispravnost rada sustava. Ako sustav radi ispravno, obrazložiti odgovor. Ako sustav ne radi ispravno, obrazložiti problem i predložiti mjere za njegovo uklanjanje (odgovor upisati u prazno polje ispod teksta).**

22. Objasniti na koji način granularnost podjele web usluge na skup sredstava utječe na skalabilnost web poslužitelja i mreže (odgovor upisati u prazno polje ispod teksta).

Ako jedan host sadrži više usluga (Multiple Services Per Host), onda to nije skalabilno rješenje, jer bi za skaliranje trebalo dodati cijeli novi host sa svim tim uslugama. U slučaju kada jedan host sadrži samo jednu uslugu (Single Service Per Host), npr. jedan host sadrži samo bazu podataka, a drugi host sadrži samo aplikacijsku logiku, onda to jest skalabilno rješenje jer bi se prilikom skaliranja baze trebalo dodati samo host sa bazom, što znači da je skaliranje puno jednostavnije. Dakle, ako je cijela aplikacija krupno zrnata, to znači da sve usluge idu na jedan host i skaliranje je slabije. S druge strane, ako je aplikacija sitnozrnata, moguće je podijeliti usluge po različitim hostovima, što značajno olakšava skalabilnost.

23. Ispravno oblikovanim REST sučeljem (engl. REST API) smatra se ono koje je klijentima izloženo na način

`DELETE /users/1234`

24. Cjevovodni režim rada HTTP protokola (engl. HTTP pipelining) omogućava

upućivanje HTTP zahtjeva prema poslužitelju bez čekanja HTTP odgovora na prethodno upućeni zahtjev.

25. Nakon početne uspostave veze, protokol WebSocket omogućava komunikaciju klijenta i poslužitelja koja je

dvosmjerna asinkrona.

26. Nakon početne uspostave veze, tehnologija Server-Sent Events omogućava komunikaciju klijenta i poslužitelja koja je

jednosmjerna od poslužitelja prema klijentu.

27. Protokolom WebSocket moguće je prenositi

tekstualne i binarne poruke.

28. Tehnologijom Server-Sent Events moguće je prenositi

tekstualne poruke izravno, a binarne samo ako su prethodno kodirane u Base64 format.

29. U sustavu s pravilnim periodičkim promjenama gdje je period promjene dugačak (primjerice, svakih nekoliko sati), a količina korisnih podataka značajno veća od zaglavlja HTTP protokola, promjene s poslužitelja na klijente preporučljivo je izvesti tehnikom prozivanja (polling).

30. Glavna zadaća RPC sustava je

normalizacija semantike poziva funkcija u istim i različitim adresnim prostorima.

31. U sustavima zasnovanim na WS-* tehnologijskom stogu, WSDL služi kao

IDL.

- 32. Promjena trećeg broja u zapisu verzije sustava koji koristi semantičko verzioniranje označava ispravke manjih grešaka postojeće funkcionalnosti.**
- 33. Poredati vrste testova, u kontekstu mikro-usluga, od sporijih prema bržima: 1. E2E (end-to-end) testovi, 2. testovi usluga (service tests), 3. jedinični testovi (unit tests).**
1, 2, 3
- 34. Oblikovni obrazac Tolerant Reader koristi se za čitanje poruka promjenjive strukture.**
- 35. Poredati serijalizacijske formate prema veličini (prosječne) poruke, od manje prema većoj. Napomena: Protobuf i Thrift se odnose na kanonske inačice binarnih formata ugrađenih u istoimene tehnologije.**
Protobuf, JSON, XML, SOAP, Thrift.
- 36. Osnovni skup tehnologija Web Services tehnologijskog stoga (WS-*) čine SOAP, WSDL, UDDI**